UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/625,812	07/26/2000	Timothy J. Van Hook	0007057-0012/000105 B S	8263

66498 7590 03/28/2007
THELEN REID BROWN RAYSMAN & STEINER, LLP
900 THIRD AVENUE
NEW YORK, NY 10022

EXAMINER	
HSU, JONI	

ART UNIT	PAPER NUMBER
2628	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/28/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	**Application No.** 09/625,812	**Applicant(s)** VAN HOOK, TIMOTHY J.
	Examiner Joni Hsu	**Art Unit** 2628

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 22 September 2006.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-18 and 23-33* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *25-31* is/are allowed.

6)☒ Claim(s) *1-18,23,24,32 and 33* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. Applicant's arguments, see page 7, filed September 22, 2006, with respect to Claims 25-31 have been fully considered and are persuasive. The 35 U.S.C. 102(e) rejections of Claims 25-27 and the 35 U.S.C. 103(a) rejections of Claims 28-31 have been withdrawn.

2. Applicant's arguments filed September 22, 2006, with respect to Claims 1-18, 23, 24, 32, and 33 have been fully considered but they are not persuasive.

3. With regard to Claim 1, Applicant argues that even though Krishna (US006161173A) describes a goal of achieving latency of one clock cycle, it admits this is not always possible. Since there is not teaching in Krishna of how to achieve the latency of one clock cycle, it would not make it obvious to modify Joffe (US006330584B1) to achieve such latency. Applicant asks for specific teaching from Krishna on how to modify Joffe to result in the claimed invention. A mere suggestion or expression of desire for a result is not a teaching of that result. There must be specific language and structure that enables such a result (page 7).

In reply, the Examiner points out that Claim 1 recites the limitation of "having an **average** pipeline latency of one instruction per cycle". Krishna discloses that the scheduler allocates a fixed latency, which is typically one clock cycle, between issuing an instruction to an execution pipeline and the execution pipeline returning a result (Col. 4, lines 1-4). For some instructions, the execution pipeline has a longer latency (Col. 4, lines 4-5). Since the fixed latency is **typically** one clock cycle for one instruction, Krishna is considered to teach how to achieve the average latency of one clock cycle, as recited in Claim 1. Even though Joffe does not explicitly teach that the execution pipeline (Col. 1, lines 17-18) has an average pipeline latency of one instruction per cycle, Krishna teaches that it is typical for instructions in an execution pipeline to have an average pipeline latency of one instruction per cycle. Therefore, it would be obvious to one of ordinary skill in the art that the execution pipeline of Joffe can be used to execute instructions that have an average pipeline latency of one instruction per cycle.

4. With regard to Claims 14 and 23, Applicant argues that Krishna teaches idling the processor, and the present claimed invention does not allow idling of the processors during operation (page 8).

In reply, the Examiner points out that Claims 14 and 23 both recite the limitation "no no-op **or** idle is inserted". Since the word "or" is used, this limitation is interrupted to mean that either no no-op is inserted, or no idle is inserted, and only one of the conditions needs to be satisfied since the word "or" is used. Since Krishna discloses that the operation is executed if no no-op is inserted (*information in each entry describes either no-op or an associated operation which is to be executed,* Col. 5, lines 36-38; *operation pipelines,* Col. 2, lines 41-45), Krishna satisfies one of the conditions, and therefore is still considered to teach this limitation.

5. With regard to Claim 33, Applicant argues that in contrast to Joffe, there would be no reaccessing the resource (register) in Claim 33 because the claim calls out executing instructions of the first program until the program is completed (page 8).

In reply, the Examiner points out that Joffe discloses that if the wait signal is asserted, the instruction execution is not completed, so the instruction will be executed again until the wait signals are deasserted, then the next instruction can be executed (Col. 10, lines 20-24, 31-32), and the process repeats until all the instructions have been executed. Therefore, Joffe teaches executing instructions of the first program until the program is completed, as recited in Claim 33.

Applicant argues that there is not teaching in Joffe of checking to see if a second register slot is available to assign to a second program from a plurality of programs (page 8).

In reply, the Examiner disagrees. Joffe teaches that the wait signal is asserted if the register is not available, and the wait signal is deasserted if the register is available for a new instruction (Col. 10, lines 20-24, 31-32). Each task (program) has a separate register and separate flags (Col. 2, lines 11-13). Therefore, a second output register slot is assigned to a second program. If a task attempt to access an unavailable resource, the task is suspended. When the resource becomes available, the suspended task is resumed, and the instruction accessing the resource is executed (Col. 2, lines 29-34). Therefore, Joffe teaches checking to see if a second register slot is available to assign to a second program from a plurality of programs.

6. With regard to Claim 32, Applicant argues that the mere suggestion in Krishna of the advantage of achieving an average latency of one clock cycle (while admitting it is not always possible to do so) does not amount to the actual teaching of how to achieve such desired latency (pages 8-9).

In reply, the Examiner points out that Claim 32 recites the limitation "each of said instructions is **issued** to said one or more units in each cycle." Krishna discloses that the schedule presumes operations issued to a particular execution unit all have the same latency of one clock cycle, even though some of the operations have longer latencies (Col. 2, lines 38-41). Therefore, Krishna teaches that each of the instructions is issued to one unit in each cycle, even though some of the instructions have longer latencies. Since Krishna teaches that each of the instructions is issued to one unit in each cycle, Krishna is considered to teach the limitation as recited in Claim 32.

7. Applicant argues that there are multiple instances where the Examiner states that a claim is "similar in scope" to another claim. Applicant does not agree that such claims have similar scope and requests that such characterization be avoided if possible (page 9).

In reply, the Examiner has now explicitly written out the rejection for each of the claims to make the rejections more clear.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

10. Claims 1-7, 9-11, 14-17, 23, 24, and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Joffe (US006330584B1) in view of Krishna (US006161173A).

11. With regard to Claim 1, Joffe describes a programmable processor (160, Figure 1) for executing a plurality of tasks (pipelined multi-tasking processor (microcontroller) 160, Col. 3, lines 40-42; *processor executes several tasks*, Col. 2, lines 66-67; *load programs into the microcontroller for execution*, Col. 5, lines 48-51). According to the disclosure of this application, the definition of "program" is an operation performed on data. For example, operations on different data represent different programs (page 8, line 5). Joffe describes that each task is an operation performed on data (*each task processes a separate frame of data*, Col. 2, lines 41-42), and therefore the tasks are programs. The programmable processor comprises an execution pipeline (Col. 3, lines 40-42). The execution pipeline has seven stages for the seven tasks (Col. 9, line 45-Col. 10, line 41). Therefore, the execution pipeline has a depth equal to the plurality of programs. Joffe discloses an interleaver for interleaving instructions from the plurality of programs and providing the instructions to the pipeline for execution (Col. 2, lines 29-34). Joffe describes that more than one task for each data flow are provided to the pipeline,

which allows the pipeline to start processing the next frame before the processing of the previous frame in the same data flow is completed (Col. 7, lines 44-49; Col. 8, lines 8-15) and therefore the number of the plurality of programs that are interleaved is greater than or equal to the depth of the pipeline.

However, Joffe does not explicitly teach that the execution pipeline has an average pipeline latency of one instruction per cycle. However, Krishna discloses that the scheduler allocates a fixed latency, which is typically one clock cycle, between issuing an instruction to an execution pipeline and the execution pipeline returning a result (Col. 4, lines 1-4). For some instructions, the execution pipeline has a longer latency (Col. 4, lines 4-5). Since the fixed latency is typically one clock cycle for one instruction, Krishna is considered to teach that the execution pipeline has an average latency of one instruction per cycle.

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify the device of Joffe so that the execution pipeline has an average pipeline latency of one instruction per cycle as suggested by Krishna because it results in a more streamlined pipeline operation and simplified design (Krishna, Col. 2, lines 60-67). Even though Joffe does not explicitly teach that the execution pipeline (Col. 1, lines 17-18) has an average pipeline latency of one instruction per cycle, Krishna teaches that it is typical for instructions in an execution pipeline to have an average pipeline latency of one instruction per cycle. Therefore, it would be obvious to one of ordinary skill in the art that the execution pipeline of Joffe can be used to execute instructions that have an average pipeline latency of one instruction per cycle.

12. With regard to Claim 2, Joffe describes that the pipeline has a datapath with a depth equal to the number of programs (Col. 1, lines 62-65).

13. With regard to Claim 3, Joffe describes that a next instruction from one of the plurality of programs is not provided to the pipeline until a previous instruction of the one of the plurality of programs has completed (*the task does not get access to the same resource until after every other task sharing the resource has finished accessing the resource*, Col. 2, lines 35-39).

14. With regard to Claim 4, Joffe describes that each program of the plurality of programs is independent of the other of the plurality of programs (Col. 1, line 62-Col. 2, line 7).

15. With regard to Claim 5, Joffe describes interleaving instructions (Col. 2, lines 29-34), and therefore the instructions are executed out of order.

However, Joffe does not teach an output buffer for storing out of order data output. However, Krishna discloses that the execution engine (140, Figure 1) has an out-of-order architecture (Col. 5, lines 11-12), and the scheduler (150) receives results from the execution units (170, 175, 180) and stores the results (Col. 5, lines 28-35). Therefore, Krishna inherently discloses an output buffer for storing out of order data output.

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify the device of Joffe to include an output buffer for storing out of order data output as suggested by Krishna because Krishna suggests that since the instructions are executed out of order (Col. 5, lines 11-12), the output buffer is needed to store the out of order data output so that the data can put in the correct order (Col. 5, lines 28-35).

16. With regard to Claim 6, Joffe describes one or more of a register copy, program counter, and program counter stack provided for each of the plurality of programs (*separate registers for each task, each task has a separate program counter (PC)*, Col. 2, lines 8-13).

17. With regard to Claim 7, Joffe describes that one or more of control and computing resources, instructions, instruction memory, data paths, data memory, and caches are shared by the plurality of programs (*multiple tasks share one or more resources*, Col. 2, lines 35-39).

18. With regard to Claim 9, Joffe describes that the instructions comprise load instructions for loading data from a data memory (*load/store unit 330 queues load and store requests to load a register from memory or to store register contents to memory*, Col. 9, lines 35-41).

19. With regard to Claim 10, Joffe describes that the instructions comprise store instructions for storing data in a memory (Col. 9, lines 35-41).

20. With regard to Claim 11, it would have been obvious to one of ordinary skill in the art at the time of invention by applicant to have data memory comprising a cache because it provides for faster execution of programs in a processor system.

21. With regard to Claim 14, Joffe describes a method of executing instructions from a plurality of programs comprising identifying N programs of the plurality of programs (Col. 2, lines 11-14, 66-67); interleaving instructions from the N programs in a processor pipeline (160, Figure 1; Col. 2, lines 29-34; Col. 3, lines 40-42); and executing the instructions such that a first instruction from one of the N programs is completed before beginning execution of a second instruction of the one of the N programs (Col. 2, lines 35-39)

However, Joffe does not teach that the pipeline has an average latency of one instruction per cycle and checking that no no-op is inserted into the pipeline for the purpose of ensuring that the first instruction is completed before beginning execution of the second instruction. However, Krishna discloses that the scheduler allocates a fixed latency, which is typically one clock cycle, between issuing an instruction to an execution pipeline and the execution pipeline returning a result (Col. 4, lines 1-4). For some instructions, the execution pipeline has a longer latency (Col. 4, lines 4-5). Since the fixed latency is typically one clock cycle for one instruction, Krishna is considered to teach that the execution pipeline has an average latency of one instruction per cycle. This would be obvious for the same reasons given in the rejection for Claim 1. Krishna also describes that the local scheduling circuitry stops the main scheduler from issuing a selected operation if the latency of another operation would create a conflict with the main scheduler issuing the selected operation (Col. 2, lines 56-60). Therefore, it is ensured that the first instruction is completed before beginning execution of the second instruction. The operation is executed if no no-op is inserted into the pipeline (*information in each entry describes either no-op or an associated operation which is to be executed*, Col. 5, lines 36-38; *operation pipelines,* Col. 2, lines 41-45). Therefore, when no no-op is inserted into the pipeline, this ensures that the first instruction is completed before beginning execution of the second instruction.

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify the device of Joffe to include checking that no no-op is inserted into the pipeline for the purpose of ensuring that the first instruction is completed before beginning execution of the second instruction as suggested by Krishna because Krishna suggests that a no-op is needed in order to indicate that the first instruction has not yet completed (Col. 5, lines 26-38).

22. With regard to Claim 15, Joffe discloses including the step of assigning a program counter to each of the N programs (*separate registers for each task, each task has a separate program counter (PC)*, Col. 2, lines 8-13).

23. With regard to Claim 16, Joffe discloses including the step of assigning a register to each of the N programs (*separate registers for each task, each task has a separate program counter (PC)*, Col. 2, lines 8-13).

24. With regard to Claim 17, Joffe discloses that the graphics processing execution pipeline has a depth of N (Col. 1, lines 62-65).

25. With regard to Claim 23, Joffe discloses a programmable processor for executing a plurality of programs, the programmable processor comprising an execution pipeline (Col. 2,

lines 11-14, 66-67). The execution pipeline has seven stages for the seven tasks (programs) (Col. 9, line 45-Col. 10, line 41). Therefore, the execution pipeline has a depth equal to the plurality of programs. Joffe discloses an interleaver for interleaving instructions from the plurality of programs (Col. 2, lines 29-34; Col. 3, lines 40-42) and providing the instructions to the pipeline for execution wherein a next instruction from one of the plurality of programs is not provided to the pipeline until a previous instruction of the one of the plurality of programs has completed (Col. 2, lines 35-39)

However, Joffe does not teach that the execution pipeline has an average pipeline latency of one instruction per cycle and wherein no no-op is inserted into the pipeline for the purpose of ensuring that the next instruction is not provided to the pipeline until the previous instruction has completed. However, Krishna discloses that the scheduler allocates a fixed latency, which is typically one clock cycle, between issuing an instruction to an execution pipeline and the execution pipeline returning a result (Col. 4, lines 1-4). For some instructions, the execution pipeline has a longer latency (Col. 4, lines 4-5). Since the fixed latency is typically one clock cycle for one instruction, Krishna is considered to teach that the execution pipeline has an average latency of one instruction per cycle. Krishna also describes that the local scheduling circuitry stops the main scheduler from issuing a selected operation if the latency of another operation would create a conflict with the main scheduler issuing the selected operation (Col. 2, lines 56-60). Therefore, it is ensured that the first instruction is completed before beginning execution of the second instruction. The operation is executed if no no-op is inserted into the pipeline (*information in each entry describes either no-op or an associated operation which is to be executed*, Col. 5, lines 36-38; *operation pipelines*, Col. 2, lines 41-45). Therefore, when no no-op is inserted into the pipeline, this ensures that the first instruction is completed before beginning execution of the second instruction. This would be obvious for the same reasons given in the rejection for Claim 14.

26. With regard to Claim 24, Joffe discloses that the execution pipeline has seven stages for the seven tasks (programs) (Col. 9, line 45-Col. 10, line 41). Therefore, Joffe discloses a method of gathering instructions from a plurality of programs comprising identifying N programs of the plurality of programs wherein N is equal to the depth of a processor pipeline (Col. 9, line 45-Col. 10, line 41); interleaving instructions from the N programs in the processor pipeline (Col. 2, lines 29-34).

However, Joffe does not teach executing the instructions such that there is an average pipeline latency of one instruction per cycle. However, Krishna discloses that the scheduler allocates a fixed latency, which is typically one clock cycle, between issuing an instruction to an execution pipeline and the execution pipeline returning a result (Col. 4, lines 1-4). For some instructions, the execution pipeline has a longer latency (Col. 4, lines 4-5). Since the fixed latency is typically one clock cycle for one instruction, Krishna is considered to teach that the execution pipeline has an average latency of one instruction per cycle. This would be obvious for the same reasons given in the rejection for Claim 1.

27. With regard to Claim 33, Joffe describes a method of executing one or more instructions from a plurality of programs (Col. 2, lines 66-67), comprising assigning a first output register slot to a first of the plurality of programs (Col. 2, lines 8-11). If the wait signal is asserted, the

instruction execution is not completed, so the instruction will be executed again until the wait signals are deasserted, then the next instruction can be executed (Col. 10, lines 20-24, 31-32), and the process repeats until all the instructions have been executed. Therefore, Joffe teaches executing instructions of the first program until the program is completed; loading output of the first program into its reserved space when the first program is completed (Col. 9, lines 26-41); checking to see if all of the plurality of programs are completed (Col. 2, lines 35-39). Joffe teaches that the wait signal is asserted if the register is not available, and the wait signal is deasserted if the register is available for a new instruction (Col. 10, lines 20-24, 31-32). Each task (program) has a separate register and separate flags (Col. 2, lines 11-13). Therefore, a second output register slot is assigned to a second program. If a task attempt to access an unavailable resource, the task is suspended. When the resource becomes available, the suspended task is resumed, and the instruction accessing the resource is executed (Col. 2, lines 29-34). Therefore, Joffe teaches checking to see if a second register slot is available to assign to a second program from a plurality of programs when the first program is completed; checking to see if one or more instructions are available when at least one of the plurality of programs is not completed (Col. 2, lines 35-39).

However, Joffe does not teach placing a no-op when no more instructions are available or the second output register slot is not available. However, Krishna discloses information in each entry describes either no-op or an associated operation which is to be executed (Col. 5, lines 36-38). Therefore, when there is a no-op, that means that no more instructions are available. This would be obvious for the same reasons given in the rejection for Claim 14.

28. Claims 8 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Joffe (US006330584B1) and Krishna (US006161173A) in view of Nguyen (US005961628A).

29. With regard to Claim 8, Joffe and Krishna are relied upon for the teachings as discussed above relative to Claim 1. The Joffe-Krishna combination implicitly discloses SIMD execution of vector instructions without addressing vector lengths.

However, Joffe and Krishna do not explicitly disclose that the processor executes SIMD vector instructions of vector length N and executes in parallel a plurality of instructions having SIMD vector lengths that sum up to N. However, Nguyen explicitly discloses that the processor executes SIMD vector instructions of vector length N and executes in parallel a plurality of instructions having SIMD vector lengths that sum up to N (Col. 1, lines 11-24; Col. 53-60).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify the devices of Joffe and Krishna so that the processor executes SIMD vector instructions of vector length N and executes in parallel a plurality of instructions having SIMD vector lengths that sum up to N as suggested by Nguyen because it provides a way to reduce processing time for repetitive tasks (Col. 1, lines 10-25).

30. With regard to Claim 18, Joffe does not explicitly disclose that the processor executes SIMD vector instructions of vector length N and executes in parallel a plurality of instructions having SIMD vector lengths that sum up to N. However, Nguyen explicitly discloses that the processor executes SIMD vector instructions of vector length N and executes in parallel a

plurality of instructions having SIMD vector lengths that sum up to N (Col. 1, lines 11-24; Col. 53-60). This would be obvious for the same reasons given in the rejection for Claim 8.

31. Claims 12 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Joffe (US006330584B1) and Krishna (US006161173A) in view of Narayanaswami (US005973705A).

Joffe and Krishna are relied upon for the teachings as discussed above relative to Claim 9.

However, Joffe and Krishna do not disclose that address space of the data memory comprises a frame buffer unit and a texture memory unit. However, Narayanaswami discloses explicitly a SIMD graphics processing system comprising a frame buffer unit (frame buffer 110f, Figure 2A) while implicitly suggesting a texture memory unit.

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify the devices of Joffe and Krishna so that address space of the data memory comprises a frame buffer unit and a texture memory unit as suggested by Narayanaswami because it provides a way to reduce processing time (Col. 2, lines 20-22).

32. Claim 32 is rejected under 35 U.S.C. 103(a) as being unpatentable over Narayanaswami (US005973705A) in view of Krishna (US006161173A).

According to the disclosure of this application, a complex instruction includes operations such as matrix multiply, vector normalization, trigonometric functions, and exponentiation (page 21, lines 1-4). Narayanaswami discloses a method of executing one or more complex or compound instructions from a plurality of programs, comprising implementing the instructions in one or more pipelined units (Col. 2, lines 35-63).

However, Narayanaswami does not teach that each of the instructions is issued to the one or more units in each cycle. However, Krishna discloses that the schedule presumes operations issued to a particular execution unit all have the same latency of one clock cycle, even though some of the operations have longer latencies (Col. 2, lines 38-41). Therefore, Krishna teaches that each of the instructions is issued to one unit in each cycle, even though some of the instructions have longer latencies. Since Krishna teaches that each of the instructions is issued to one unit in each cycle, Krishna is considered to teach the limitation as recited in Claim 32. This would be obvious for the same reasons given in the rejection for Claim 1.

Allowable Subject Matter

33. Claims 25-31 are allowed.

The following is a statement of reasons for the indication of allowable subject matter:

34. The prior art taken singly or in combination do not teach or suggest a programmable processor comprising **a target program counter coupled to a plurality of program counters, the target program counter for determining a number of programs to interleave from a plurality of programs that is greater than the number of program counters**; each of the **plurality of program counters** coupled to an instruction memory; instructions from the instruction memory coupled to an instruction decode; the decode coupled to a plurality of registers; each of the plurality of registers coupled to an operand route; the operand route

coupled to an arithmetic datapath; the data path and an output of a data memory coupled to a result route; and an output of the result route fed back to each of the plurality of registers, as recited in Claim 25. Claims 26-31 depend from Claim 25, and therefore also contain allowable subject matter.

35. The closest prior art (Joffe US006330584B1) teaches a programmable processor (160, Figure 1) for executing a plurality of programs (pipelined multi-tasking processor (microcontroller) 160, Col. 3, lines 40-42; *processor executes several tasks*, Col. 2, lines 66-67; *load programs into the microcontroller for execution*, Col. 5, lines 48-51). According to the disclosure of this application, the definition of "program" is an operation performed on data. For example, operations on different data represent different programs (page 8, line 5). Joffe describes that each task is an operation performed on data (*each task processes a separate frame of data*, Col. 2, lines 41-42), and therefore the tasks are programs. The programmable processor comprises a task control block (320, Figure 5) coupled to a plurality of registers 315 (Col. 9, lines 56-61; Col. 2, lines 11-13); each of the plurality of registers 315coupled to an instruction memory (314; Col. 10, lines 1-5); instructions from the instruction memory coupled to an instruction decode (Col. 10, lines 1-7); the decode coupled to a plurality of registers 312 (Col. 10, lines 6-12); each of the plurality of registers 312 coupled to an operand route; the operand route coupled to an arithmetic datapath (318; Col. 10, lines 8-12); the datapath and an output of a data memory (312) coupled to a result route; and an output of the result route fed back to each of the plurality of registers 312 (Col. 9, lines 26-28), as shown in Figure 5. However, Joffe does

not teach a target program counter coupled to a plurality of program counters, the target program counter for determining programs to interleave.

36. Another prior art (Nemirovsky US005115513A) teaches two program counters for instructions that are interleaved (*two processors having separate program counter, instructions are interleaved so that as one processor fetches instructions the other executes*, Abstract; *to carry out the pipeline interleaving technique the control unit 22 defines the two machines*, Col. 5, lines 27-29; *program counter for the single instruction machine 62 and a program counter for the multiple instruction machine 64 comprise 12 presettable counters with incrementing ability which access the microstore 48 through a multiplexer 66 which alternates between the SIM-PC 62 and the MIM-PC 64, multiplexer toggle for each instruction period to switch between the two machines*, Col. 6, lines 34-44; *concurrent execution of two or more programs each independent of the other*, Col. 2, lines 42-45). However, Nemirovsky does not teach a target program counter coupled to a plurality of program counters, the target program counter for determining a number of programs to interleave from a plurality of programs that is greater than the number of program counters.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The following prior art teach SIMD processing and execution of pipelines in superscalar processors.

U.S. Patent No. 6,470,445 B1 to Arnold et al. U.S. Patent No. 5,420,990 to McKeen et al.

U.S. Patent No. 6,064,818 to Brown et al.

U.S. Patent No. 6,282,635 to Sachs

U.S. Patent No. 5,949,996 to Atsushi

U.S. Patent No. 5,548,737 to Edrington et al.

U.S. Patent No. 5,809,552 to Kuroiwa et al.

U.S. Patent No. 5,115,513 to Nemirovsky et al.

U.S. Patent No. 5,428,807 to McKeen et al.

U.S. Patent No. 5,802,386 to Kahle at al.

U.S. Patent No. 6,209,078 to Chiang et al.

U.S. Patent No. 6,412,061 to Dye

U.S. Patent No. 6,508,862 to Joy et al.

In particular, U.S. Patent No. 6,507,862 to Joy et al. discloses vertical and horizontal threaded processors. Joy et al. discloses a single pipeline shared among a plurality of machine states or threads, a thread that is currently active, not stalled, is selected and supplied data or functional blocks connected to the pipeline; when active thread is stalled, the pipeline immediately switches to a non-stalled thread, and begins executing the non-stalled thread (Col. 6, lines 10-40; Col. 8, lines 15-60).

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joni Hsu whose telephone number is 571-272-7785. The examiner can normally be reached on M-F 8am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ulka Chauhan can be reached on 571-272-7782. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JH

[illegible]

KEE M. TUNG
SUPERVISORY PATENT EXAMINER